

Compte Rendu

Traveaux pratiques java embarqué

**Réalisé par :**

Insaf Arab N°5

**Encadré par :**

Pr. Youness Khamlichi

2eme année cycle d’ingénieur

Génie systèmes embarqués et informatique industrielle

# **Exercice 1**

**Hello world**

La première application est une MIDlet basique. Elle crée une classe `PrintMIDlet` qui étend `MIDlet`, une classe de base pour les applications MIDP (Mobile Information Device Profile).

L'application crée un formulaire (`Form`) nommé "Bonjour Midlet" dans le constructeur de la classe. Lorsque l'application démarre (`startApp()`), elle obtient l'instance d'affichage (`Display`) et défini le formulaire comme l'affichage courant. Les méthodes `pauseApp()` et `destroyApp()` sont laissées vides, ce qui est courant pour les applications simples qui n'ont pas besoin de logique spécifique pour ces états.

import javax.microedition.lcdui.\*;

import javax.microedition.midlet.\*;

public class PrintMIDlet extends MIDlet {

    private Display display;

    private Form helloForm;

    public PrintMIDlet() {

        helloForm = new Form("Bonjour Midlet");

    }

    protected *void* startApp() {

        display = Display.getDisplay(this);

        display.setCurrent(helloForm);

    }

    protected *void* pauseApp() {}

    protected *void* destroyApp(*boolean* *unconditional*) {}

}

****

# **Exercice 2**

**Formulaire**

Ce code Java crée une application MIDlet simple qui affiche un formulaire avec des éléments statiques tels que du texte ("Hello", "GSEII2", etc.) et deux champs pour saisir le nom et l'e-mail. L'application utilise `CommandListener` pour détecter la commande de sortie et arrêter l'application lorsqu'elle est invoquée.

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class FormMidlet extends MIDlet implements CommandListener {

    private Display display;

    private Form form;

    private *boolean* started = false;

    public FormMidlet() {

    }

    protected *void* startApp() {

        if (!started) {

            display = Display.getDisplay(this);

            form = new Form("Item Layout");

            form.append("Hello");

            form.append("GSEII2");

            form.append("\nThis is a Form");

            form.append("\nEnter your information");

            form.append(new TextField("Name", null, 32, TextField.ANY));

            form.append(new TextField("Email", null, 32, TextField.ANY));

            display.setCurrent(form);

            started = true;

        }

    }

    protected *void* pauseApp() {}

    protected *void* destroyApp(*boolean* *unconditional*) {}

    public *void* commandAction(Command *c*, Displayable *d*) {

        if (*c*.getCommandType() == Command.EXIT) {

            destroyApp(true);

            notifyDestroyed();

        }

    }

}



# **Exercice 3**

**Liste**

Ce code Java crée une application MIDlet qui affiche une liste déroulante avec des éléments pré-définis comme "Item1", "Item2", etc., permettant la sélection de plusieurs éléments à la fois. L'application utilise les classes de base de MIDP pour gérer l'interface utilisateur et ne contient pas de logique spécifique pour les événements de pause ou de destruction de l'application.

import java.util.\*;

import javax.microedition.lcdui.\*;

import javax.microedition.midlet.\*;

public class MIDletList extends MIDlet {

    private Display display;

    private List liste;

    protected static final String[] items = { "Item1", "Item2", "Item3", "Item4" };

    public MIDletList() {

        this.liste = new List("Selection", List.MULTIPLE, items, null);

    }

    protected *void* startApp() {

        this.display = Display.getDisplay(this);

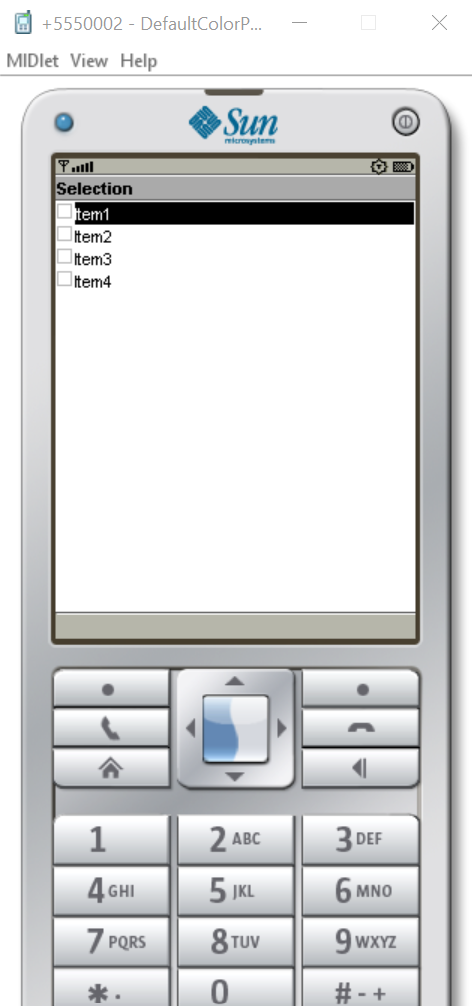
        display.setCurrent(liste);

    }

    protected *void* pauseApp() {}

    protected *void* destroyApp(*boolean* *unconditional*) {}

}



# **Exercice 4**

**Calculatrice**

Ce code Java représente une application MIDlet agissant comme une calculatrice simple avec des opérations de base (addition, soustraction, multiplication, division et modulo). Elle offre une interface utilisateur avec deux champs pour entrer des nombres et un champ pour afficher le résultat. L'utilisateur peut sélectionner l'opération désirée à partir d'une liste de boutons affichés à l'écran. L'application utilise des commandes pour gérer les opérations et la sortie de l'application.

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class Calculatrice extends MIDlet implements CommandListener{

    private Command exitCommand; // The exit command

    private Display display;     // The display for this MIDlet

    private Command add,sub,mul,div,mod;

    private TextField t1,t2,t3;

    private Form form;

    public Calculatrice() {

        display = Display.getDisplay(this);

        exitCommand = new Command("Exit", Command.EXIT,30);

        add = new Command("Addition", Command.SCREEN, 1);

        sub = new Command("Substraction", Command.SCREEN, 2);

        mul = new Command("Multiplication", Command.SCREEN, 3);

        div = new Command("Division", Command.SCREEN, 4);

        mod = new Command("Modulo", Command.SCREEN, 5);

        form = new Form("");

        form.addCommand(add);

        form.addCommand(sub);

        form.addCommand(mul);

        form.addCommand(div);

        form.addCommand(mod);

        form.addCommand(exitCommand);

        t1 = new TextField("Number1",null,30,TextField.ANY);

        t2 = new TextField("Number2",null,30,TextField.ANY);

        t3 = new TextField("Answer",null,30,TextField.ANY);

        form.append(t1);

        form.append(t2);

        form.append(t3);

        form.setCommandListener(this);

      }

    public *void* startApp() {

        TextBox t = new TextBox("Calculatrice", "Hello, World!", 256, 0);

        display.setCurrent(form);

    }

    public *void* pauseApp() {

    }

    public *void* destroyApp(*boolean* *unconditional*) {

    }

    public *void* commandAction(Command *c*, Displayable *s*) {

*double* sum;

        if (*c* == exitCommand) {

            destroyApp(false);

            notifyDestroyed();

        }

        if (*c*==add)

        {

*double* x = Double.parseDouble(("" + t1.getString()));

*double* y = Double.parseDouble(("" + t2.getString()));

             sum = x+y;

             t3.setString(""+sum);

        }

        if (*c*==sub)

        {

*double* x = Double.parseDouble(("" + t1.getString()));

*double* y = Double.parseDouble(("" + t2.getString()));

             sum = x-y;

             t3.setString(""+sum);

        }

        if (*c*==mul)

        {

*double* x = Double.parseDouble(("" + t1.getString()));

*double* y = Double.parseDouble(("" + t2.getString()));

             sum = x\*y;

             t3.setString(""+sum);

        }

        if (*c*==div)

        {

*double* x = Double.parseDouble(("" + t1.getString()));

*double* y = Double.parseDouble(("" + t2.getString()));

             sum = x/y;

             t3.setString(""+sum);

        }

        if (*c*==mod)

        {

*double* x = Double.parseDouble(("" + t1.getString()));

*double* y = Double.parseDouble(("" + t2.getString()));

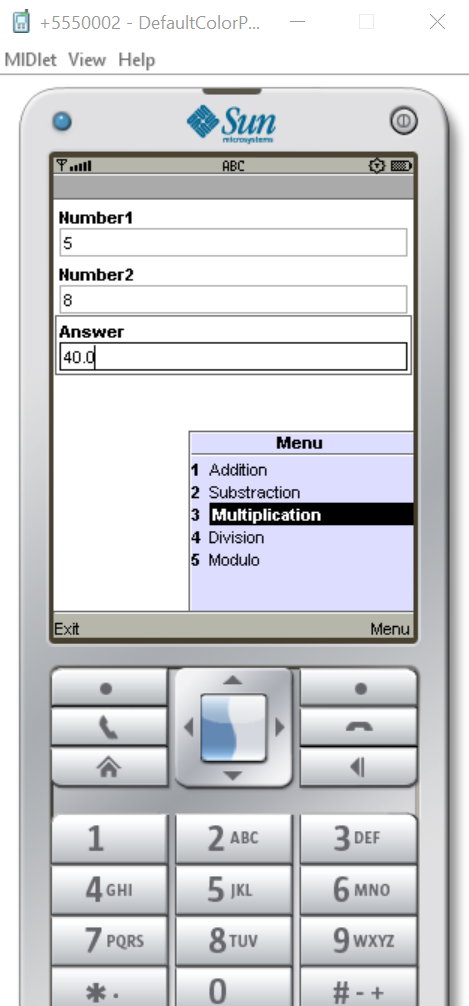
             sum = x%y;

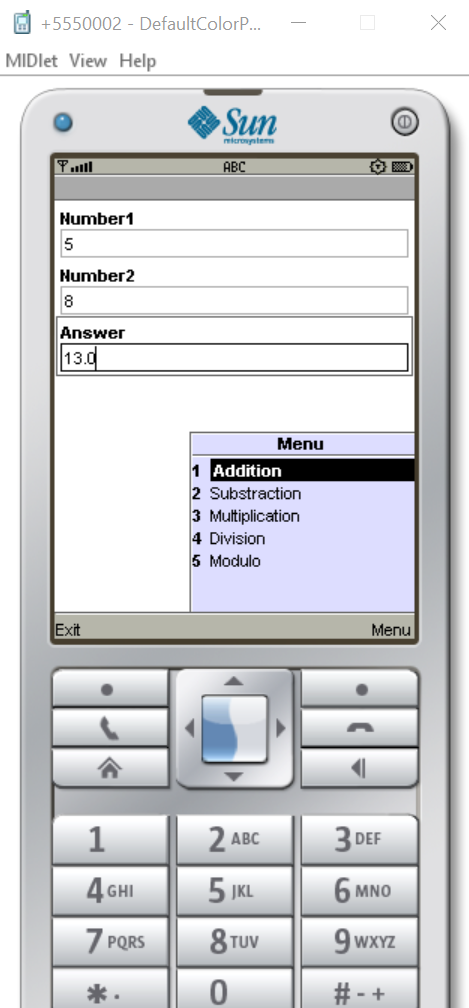
             t3.setString(""+sum);

        }

    }

}





# **Exercice 5**

**Application**

Ce code Java crée une application MIDlet représentant une plateforme d'apprentissage. Elle affiche un formulaire avec un message de bienvenue, une liste de choix, un sélecteur de date, un champ de texte pour le nom et une barre de progression pour suivre l'avancement. L'application utilise une commande "Exit" pour quitter l'application lorsque sélectionnée.

import javax.microedition.lcdui.\*;

import javax.microedition.midlet.\*;

public class Calendar extends MIDlet implements CommandListener {

    private Display display;

    private Form form;

    private *boolean* started = false;

    public Calendar() {

    protected *void* startApp() {

        if (!started) {

            display = Display.getDisplay(this);

            form = new Form("Learning Platform");

            // Text

            form.append("Welcome to your learning journey!");

            // List of Choices

            String[] choices = {"Embedded C", "Model Based Design", "Automation with Python"};

            ChoiceGroup choiceGroup = new ChoiceGroup("Choose an option:", ChoiceGroup.EXCLUSIVE, choices, null);

            form.append(choiceGroup);

            // Date

            DateField dateField = new DateField("Select a date:", DateField.DATE);

            form.append(dateField);

            // Text Field

            form.append(new TextField("Enter your name:", null, 32, TextField.ANY));

            // Progress Bar

            Gauge gauge = new Gauge("Progress:", true, 100, 0);

            form.append(gauge);

            // Exit Command

            Command exitCommand = new Command("Exit", Command.EXIT, 0);

            form.addCommand(exitCommand);

            form.setCommandListener(this);

            display.setCurrent(form);

            started = true;

        }

    }

    protected *void* pauseApp() {}

    protected *void* destroyApp(*boolean* *unconditional*) {

        // Clean up resources, if any

    }

    public *void* commandAction(Command *c*, Displayable *d*) {

        if (*c*.getCommandType() == Command.EXIT) {

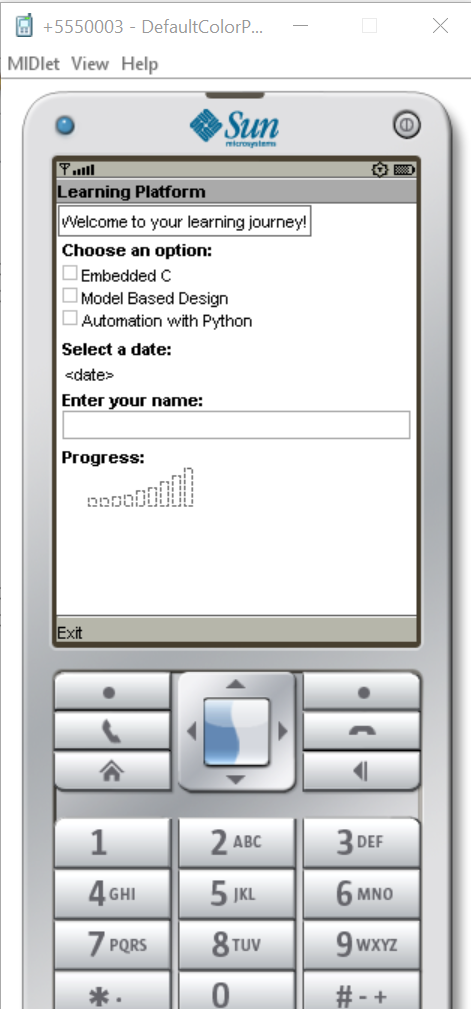
            destroyApp(true);

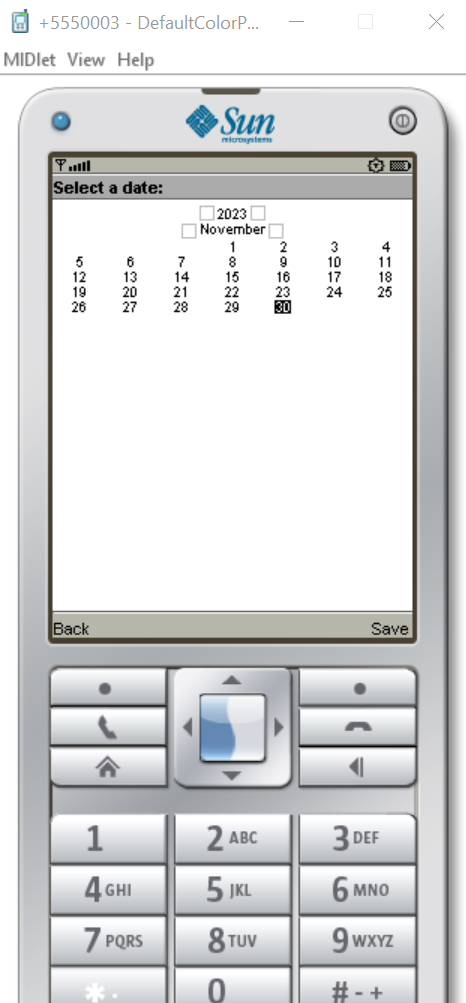
            notifyDestroyed();

        }

    }

}





# **Exercice 6**

**Image**

Ce code Java crée une application MIDlet qui affiche une image d'un coucher de soleil dans un formulaire. L'application utilise les classes Form, StringItem, Ticker, et ImageItem pour construire l'interface utilisateur.

Lorsque l'application démarre, elle crée un formulaire avec un message "Sunset View" et affiche une image d'un coucher de soleil à l'intérieur du formulaire. De plus, un défilement de texte avec le message "What a beautiful sunset !" est ajouté en tant que ticker au formulaire pour animer le contenu.

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class ImgMidlet extends MIDlet {

    private Display display;

    private Form mainForm;

    private StringItem msg;

    private Ticker t;

    public ImgMidlet(){

        display=Display.getDisplay(this);

        mainForm=new Form ("Mon formulaire");

        t=new Ticker("What a beautiful sunset !");

        msg=new StringItem("", "Sunset View");

        mainForm.append(msg);

        try{

            Image img=Image.createImage("/images/sunset.jpg");

            mainForm.append(new ImageItem(null,img,ImageItem.LAYOUT\_CENTER,null));

        } catch(java*.*io*.*IOException e) {e.printStackTrace();}

    }

    public *void* startApp(){

        mainForm.setTicker(t);

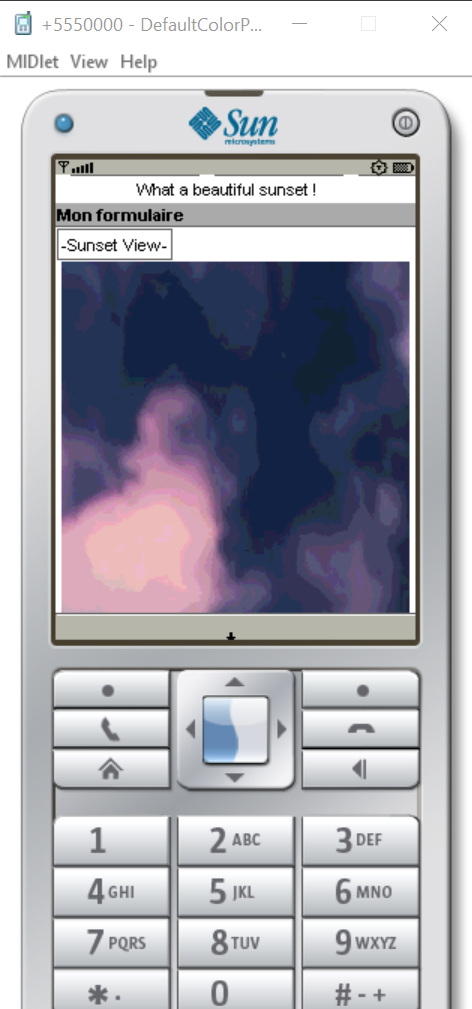
        display.setCurrent(mainForm);

    }

    protected *void* pauseApp() {}

    protected *void* destroyApp(*boolean* *unconditional*) {}

}



# **Exercice 7**

**Date & Time**

Ce code Java crée une application MIDlet affichant l'heure et la date actuelles dans une TextBox intitulée "Time". L'application utilise les classes TextBox, Date, Command, et Display pour construire l'interface utilisateur et gérer les commandes.

Lorsque l'application démarre, elle récupère la date et l'heure actuelles à l'aide de la classe Date et les affiche dans une TextBox. Deux commandes sont disponibles : "Exit" pour quitter l'application et "Help" pour afficher un message dans la console lorsque sélectionné.

Time import java.util.\*;

import javax.microedition.midlet.\*;

import javax.microedition.lcdui.\*;

public class Event extends MIDlet implements CommandListener{

    private Command exitCommand;

    private Command helpCommand;

    private Display display;

    public Event(){

        display=Display.getDisplay(this);

        exitCommand=new Command("Exit",Command.EXIT,0);

        helpCommand=new Command("Help",Command.HELP,0);

    }

    public *void* startApp(){

    String str=null;

    Date date=new Date();

    str="Date et Heure: " + date.toString();

    TextBox t=new TextBox("Time",str,256,0);

    t.addCommand(exitCommand);

    t.addCommand(helpCommand);

    t.setCommandListener(this);

    display.setCurrent(t);

    }

    public *void* pauseApp(){}

    public *void* destroyApp(*boolean* *unconditional*){}

    public *void* commandAction(*Command* *c*, *Displayable* *s*){

        if(c==helpCommand){

        Sys*t*em.out.println("help");

        }

        else if(c==exitCommand){

            dest*r*oyApp(false);

            notifyDestroyed();

        }

    }

}

